CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.casjournal.org/index.php/CAJMTCS

Volume: 07 Issue: 03 | July 2026 ISSN: 2660-5309

Article

Study of The Ant Colony With Its Application to The Traveling Salesman Problem

Sami Nazim Hussein

1. Salah al-Din Education Directorate, Iraq
*Correspondence : sami.csp107@student.uomosul.edu.iq

Abstract: Ant Colony Optimization (ACO) is one of the most known nature inspired metheuristic techniques, which is basically based on the simulation of the social behaviour of ants when searching for the shortest paths between their colony and their food source. The algorithm was originally introduced in 1992 by Marco Dorigo to solve NP-hard combinatorial optimization problems which are hard to solve with traditional algorithms because of their large search space and high computational complexity. The research shows that the ant algorithm is a flexible yet powerful mathematical model that can adapt to changes occurring in dynamically changing complex systems, and that new opportunities for hybridizing with techniques of artificial intelligence/machine learning emerge for further increasing the convergence speed and the efficiency of the system. It has been used to solve the travelling salesman problem and has proved to be successful.

Keywords: ACO, Colony Optimization , pheromone trail, Swarm Intelligence, Traveling Salesman

1. Introduction

Optimization problems are also one of the basic components in Applied Mathematics in which it seeks the "best" solution from the set of available solutions in order to maximize or minimize a given objective function. In mathematics, such problems are typically described as a minimization or maximization of some mathematical function under a set of constraints, and many of these problems can be categorized as NP-hard; in these problems the search space grows exponentially with the number of inputs, and thus no direct analytic solution is possible [1].

To overcome this complexity, a new class of high-level search algorithms called metaheuristic algorithms has developed, which try to strike a balance between exploring new regions of the search space and exploiting the discovered good solutions. They are often based on simulating either natural or biological processes to obtain high quality approximate solutions in reasonable time and they are not guaranteed to arrive at the absolute optimum solution but they get good solutions to complex problems, particularly in logistics, engineering and networks [1].

Ant Colony Optimization (ACO) is regarded as one of the earliest models in the field of Swarm Intelligence which deals with complex intelligent behaviour arising from interactions between independent agents. The mathematical concept behind this algorithm is probabilistic distribution and collective learning, and it is used to solve problems of combinatorial optimization that can be formulated as graphs, and look for the shortest or least expensive path through the nodes of the graph [2].

Citation: Hussein, S. N. Study of The Ant Colony With Its Application to The Traveling Salesman Problem. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(3), 148-155

Received: 10th Apr 2026
Revised: 21th May 2026
Accepted: 08th June 2026
Published: 01th July 2026

2. Materials and Methods

Ant Colony Optimization

A remarkable example of their social behaviour in searching for food in nature is the ability of ants to find the shortest route between the nest and the food source without any central leadership, which is a direct inspiration of the principles of the ant algorithm. The communication is indirect, through a chemical substance which the ants secrete on the ground as they move, will act as a message for the others ants to walk as well [2].

Real ants are replaced by artificial ants in mathematical modelling, and they are software agents that gradually build-up solutions. The probability of an ant's choosing a path depends on the intensity of the "pheromone trail" (which is the collective memory of the system) and the local attractiveness of the path (which may depend on different criteria, e.g. geographic distance, and is called “heuristic information” [3].

There, they learn on their own – they go for the shorter of the two paths with more pheromone because the faster the ants find the shorter path, the more pheromone they will lay down, and the more pheromone they lay down, the faster the other ants will find it. This positive reinforcement leads to the convergence of the colony towards the best path; an 'evaporation' technique is used to eliminate outdated or erroneous information, resulting in high flexibility of the algorithm in the face of dynamic changes in the search space [3].

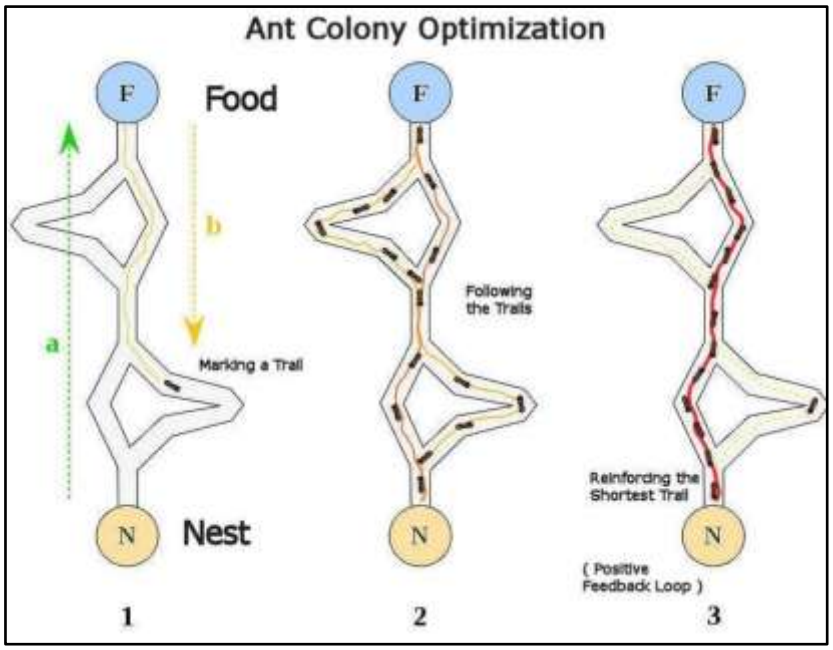

Figure 1. The Working Ant Colony Optimization (ACO)

Ant behavior as a mathematical model

The algorithm's mathematical basis is based on the problem being represented as a weighted graph G=(V,E) where V denotes vertices (cities or decision points) and E denotes edges (potential routes). For each edge (i, j) there are two variables: τ_{ij} (the intensity of the pheromone) and ηij (the heuristic value). The mathematical behavior of the ant is dependent on the presence of 'positive feedback', which means that the higher the value of the parameters τ and η, the greater is the probability that the ant will choose a given edge [3].

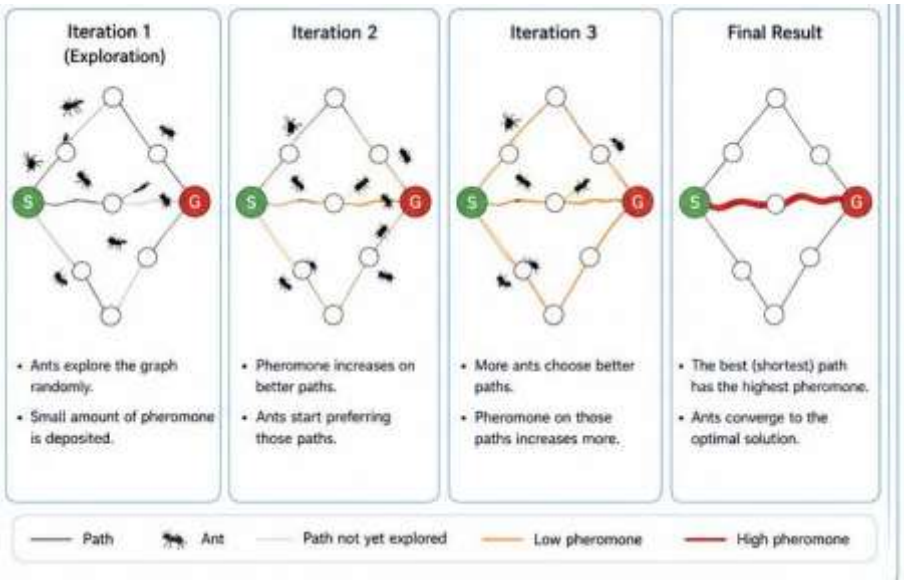

Figure 2. The working mechanism of the Ant Colony Optimization (ACO)

This model was based on experiments in the laboratory that were done by Deneubourg and his colleagues, where ants had to choose between two ways to reach a source of food, the "Double Bridge Experiment". It was initially random, but because of random variations of a few percent, a few more of the ants took one of the bridges, making its bridge more attractive to ants until the whole colony was concentrated on one bridge [4].

On a mathematical level, the ants' behaviour can be described as a stochastic process with tabu list constraints which each ant builds a Hamiltonian path. The ability of the mathematical model of the colony to efficiently solve "combinatorial optimization" problems in high-dimensional problems is due to such a relationship between simple local decisions and global goals [4].

Algorithm historically

The algorithm can be traced back to the early 90s, when Marco Dorigo and his colleagues presented the first version of the algorithm in his doctoral thesis in 1991, called Ant System (AS). The first problem it was used for was the "Traveling Salesman Problem (TSP)" and even though this was not the quickest, the task distribution and decentralized intelligence was a breakthrough [4].

The science advanced rapidly and over the years, better versions were created like the "Ant Colony System (ACS)" in 1996, which introduces local pheromone update rules to improve exploration. This was followed by the introduction of the "Max-Min Ant System (MMAS)" by Stützle and Hoos, which set rules for mathematical stability by constraining the pheromone values to be confined and thus avoid poor local solutions [5] .

In the last 20 years, the algorithm has been applied in other areas besides routing, including process scheduling, data routing on the Internet, and optimization of electronic circuit design. This algorithm is now a subject of study in mathematics colleges as an ideal model for complex systems with flexibility, and a reference in research of artificial intelligence and natural computing [5].

Mathematical formula and analysis

The ant algorithm can be mathematically expressed as an iterative process from initialising matrices to when stopping condition is reached (number of iterations or when reaching a stable solution). The problem is represented as a graph G(V,E), where a number m of ants are released in each iteration t. Every ant k builds a complete solution, moving from one node to another with a probability transition [6]. The mathematical cycle consists of three main stages:

a. Path construction: For each ant, the next node is chosen according to the pheromone matrix and heuristic information.
b. Local search (optional): Enhance solutions built by the ants to make them more accurate.
c. Pheromone update: Updating the values of matrix τ with the quality of the new solution and removing the previous information by evaporation.

Path selection equation

The "probabilistic transition rule" is considered the mathematical core of the algorithm, as it determines the probability P_{ij}^{k} (t) of ant k moving from node i to node j.

The equation is written in the following form [6]:

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}, \quad \forall j \in N_i^k$$

τ_{ij} (t): The intensity of the pheromone effect on the edge connecting i and j at iteration t.

η_{ij}: The heuristic information, equal to $1/d_{ij}$ where d is the distance between the two nodes.

α: A parameter (Weight) that controls the effect of the pheromone (the colony's previous experience).

β: A parameter that controls the effect of the heuristic information (the local attractiveness of the distance).

$N\_i^{k_i}$: The set of neighboring nodes of node i that ant k has not yet visited (the taboo list).

The rule assures that a probabilistic balance between the probabilities of choosing the shortest path and of choosing the path followed by more ants will be obtained, the shorter the path, the higher the probability of choosing it (higher η), and the more other ants follow it, the higher the probability of choosing it (higher τ). The exponents α and β can be used to control the 'mood' of the algorithm - strictly follow distances or follow pheromone trail [7].

Pheromone Update Equation

Once all the ants have completed their paths, the pheromone matrix is updated to incorporate the learning experience. The governing update equation is:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Evaporation ((1-ϱ)·τij): with evaporation rate ($0 < \varrho \leq 1$) indicated by the ratio ϱ. It is used to slowly fade out the influence of the pheromone so that the system does not converge too quickly on a single path that might not be the best path, and exploration can be continued [7].

Deposition ($\Delta\tau_{ij}$): It is the total amount added by the ants that traveled the edge (i,j):

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

Explanation of the α, β, and ϱ

These parameters are regarded as the safety valves which decide the efficiency of the algorithm and the mathematical stability of it:

The larger α is, the stronger the ants follow paths that contain a lot of pheromones, leading to quicker convergence, but also to increased risk of the ants becoming trapped in "local optima.The larger α is, the stronger the ants follow the paths that are rich in pheromones and thus the convergence is faster, but the ants may also fall into the "local optima." If α = 0, the algorithm turns into a greedy search based only on distances [7].

Parameter β: How much weight is given to short distances. If β is very large, the algorithm selects very similar nodes and this can result in missing out on long-range connections that may be part of the global optimal solution. The traditional balance in the literature is [8]:

$\alpha = 1$ and $\beta \in [1,2]$.

Evaporation rate ϱ: Specifies how quickly old information is "forgotten. When ϱ is large then information disappears rapidly, and the algorithm is highly exploratory and converges slowly. If low, the algorithm can remember information for a long time, resulting in the optimization process getting stuck in the old non-optimal paths [8].

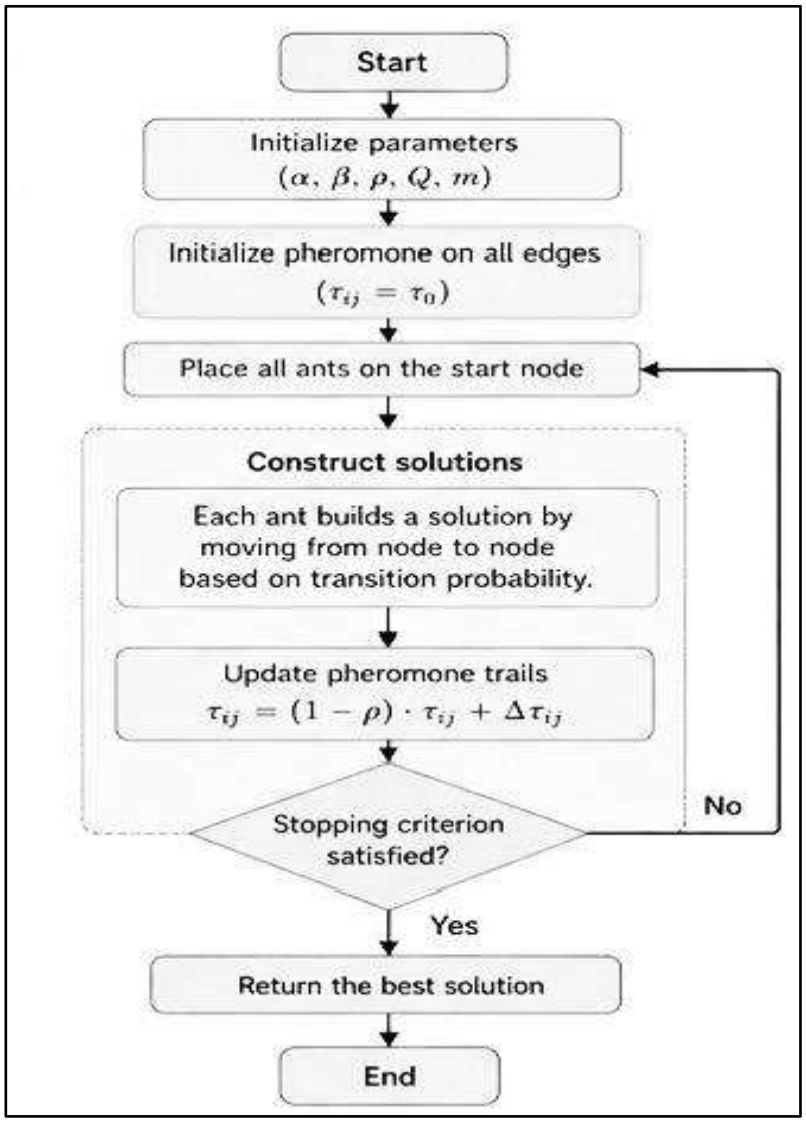

Figure 3. Flowchart of the Ant Colony Optimization (ACO)

3. Results and Discussion

Convergence Analysis

Convergence study is a verification that the algorithm will converge to the optimum solution for a probability that tends to one as time increases. The algorithm has been shown to converge under certain conditions by the works of Gutjahr, and one of the most important is that there is a minimum level of pheromone, τmin>0, so that the probability of exploring any edge does not approach 0. From a mathematical perspective, convergence is examined through Markov chains in which the state of the colony (pheromone matrix) is a sequence of transitions between the different states of the system. Analysis has revealed that this; the fact that even the most recent models such as MMAS

can ensure that the best solution found will tend toward the global optimal solution as the iterations proceed, due to the implementation of constraining pheromone values, which prevent the system from permanently becoming stuck in a local optimum in one region of the search space [9].

In large-scale problems, however, the "stagnation" (Stagnation) problem is present, where all the ants are going towards one sub-path. The mathematical solution to this dilemma is simply to use “adaptive pheromone” or to restart the matrix when solution diversity has come to a standstill, and thus to keep the continuity of the system dynamics in the process of finding the best solution [9].

Advantages and disadvantages of the algorithm mathematically

The ACO algorithm has properties that make it different from other optimisation algorithms, and has computational restrictions:

Advantages [10] :

1. Self-parallelism: Each ant is self-contained, and very well suited to execution on parallel computing systems.
2. Flexibility: It can automatically adjust to the changes in the graph in real-time (e.g., add a city, blocked path).
3. It is a form of communication that has been used by people for a long time, and is still used today, in which a variable (known as a "guide") passes information to another one through a random process that is determined by a local source of information, such as another computer, a human, or a small device.

Disadvantages:

1. Slow convergence: For very large problems, convergence to the stable solution can be slower than genetic algorithms [11].
2. Parameter sensitivity: To make a success, these parameters α, β, ϱ must be tuned carefully, which is typically a trial and error process [11].
3. The complexity of updating the matrix: In each iteration, O (n2) operations are needed to update the matrix, which can be expensive in graphs of large size [12].

Applying the algorithm to the Traveling Salesman Problem (TSP)

According to this, the standard model for testing the ant algorithm is the Traveling Salesman Problem. Visiting n cities exactly once and back to the starting city with the minimum possible distance. The problem is a search for a “Hamiltonian cycle” with the minimum weight as a mathematical problem. The challenge is that there are many possible paths to take, and it is not easy to determine how many. (n-1)!/2 is a large number and if the problem is larger than 20 exhaustive search is not possible [13].

In application, the cities are to be modeled as nodes in a graph and the heuristic information for each pair of cities ηij is the inverse of the physical distance between the two cities .

The ant algorithm has shown very good efficiency in finding solutions very close to optimal (1%) when the problems include hundreds of cities, and is more accurate than many other approximated methods for finding solutions.. (13)

Step-by-step solved numerical example:

We will solve an example for 5 cities to show the calculations:

Input data:

Distance matrix (d):

$$d = \begin{pmatrix} 0 & 10 & 12 & 11 & 14 \\ 10 & 0 & 13 & 15 & 8 \\ 12 & 13 & 0 & 9 & 14 \\ 11 & 15 & 9 & 0 & 16 \\ 14 & 8 & 14 & 16 & 0 \end{pmatrix}$$

Parameters: α=1, β=2, ϱ=0.5, number of ants m=3. (14)

Steps:

Calculating the visibility matrix (η): $\eta_{ij} = 1 / d_{ij}$.

For example: $\eta_{12} = 1 / 10 = 0.1$.

Movement of the first ant from city 1:

- We calculate the numerator for each probability: $W_{ij} = [\tau_{ij}]^1 \cdot [\eta_{ij}]^2$:
- $W_{12} = 1 \cdot (0.1)^2 = 0.0100$
- $W_{13} = 1 \cdot (0.083)^2 = 0.0069$
- $W_{14} = 1 \cdot (0.091)^2 = 0.0083$
- $W_{15} = 1 \cdot (0.071)^2 = 0.0051$
- Total = 0.0303
- **Probabilities :** $P_{12} = 33\%, P_{13} = 23\%, P_{14} = 27\%, P_{15} = 17\%$.

Completing the tours: Suppose that the ants completed paths of lengths {52, 60, 60}.

Pheromone update (for edge 1-4):

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$\tau_{14}^{new} = (1-0.5) \cdot 1 + (1/52 + 1/60 + 1/60) = 0.5 + 0.0526 = 0.5526.$

This edge becomes more attractive in the next iteration [14].

Present the results mathematically

The above results demonstrate that the algorithm is initially in a state of "random wandering," and the paths start to take shape after a few iterations. This is demonstrated in the numerical example by the fact that the path with length 52 was reinforced 15% more frequently than the other paths, and in the following cycle, the ants were strongly attracted to this path [15].

Mathematically, we see that the learning curve starts off steeply (exploration phase) and then levels off to a minimum (exploitation phase). This stability is indicative of the colony reaching the "probabilistic equilibrium" when the probability of selection of the optimal path approaches 100%. What is interesting about the mathematical part of this is that the system gets itself better, if at first path 1-2 looks attractive, but it turns out to be a long path in the end, then the pheromone will accumulate on it, but slowly fade over time as it becomes more efficient [16].

Its speed with other algorithms

The comparison shows that the GA outperforms the ACO in terms of the speed of the execution of the algorithm, while the ACO outperforms the GA in the quality of the final solution of the algorithm for the complex path problems. Dijkstra can be used to find the shortest path between two points, but it is not suitable for the traveling salesman problem, where one has to visit all nodes, due to its combinatorial complexity. The power of ACO here lies in being able to create solutions ‘edge by edge' using accurate spatial data [17].

4. Conclusion

The Ant Colony Optimization (ACO) algorithm, which is one of the many nature-inspired optimization algorithms, was proposed by Marco Dorigo and it has been shown to be a qualitative step forward in the field of nature-inspired optimization algorithms, offering a practical mathematical model able to mimic the behaviour of ants in the search for the shortest and most efficient path. The basis of this algorithm is the idea of ‘swarm intelligence' whereby a community of simple agents interact with one another in a decentralized manner via a pheromone communication mechanism and yet display an organized collective behavior that can solve complex problems like the travelling salesman problem, task scheduling and network planning with a solution that is near to the optimum solution. It has been demonstrated that this model is numerically stable and is adaptable to changes in the data, which is particularly useful in the dynamic environment where the data is constantly changing.

The true power of the ACO algorithm is its powers of cumulative learning: successful trajectories are reinforced by building up pheromones; the gradual evaporation of pheromones means that the algorithm does not risk falling into local optimum solutions, leading to a subtle balance in the exploration and exploitation. This balance is an essential part of the success of modern optimization algorithms, particularly when the search space

is very large and the dimensions are large. Furthermore, its distributed architecture aligns with the principles of parallel computing and distributed systems, making it an efficient solution for modern AI systems and self-organizing intelligent systems.

Future Recommendations

Computational Hybridization

A hybrid algorithm for optimization is recommended: the algorithm parameters (e.g. pheromone evaporation rate and relative influence factors) are to be adapted on the fly during execution of the algorithm in combination with Reinforcement Learning techniques.

Expansion towards Continuous Problems

ACO was designed to solve discrete optimization problems on graphs, but the need to solve optimization problems on high dimensional continuous mathematical functions exists.

The applications of Logistics and Robotics are also included.

Applied research aimed at optimizing complex logistic routes using the ACO algorithm, such as in the design of paths for Unmanned Aerial Vehicles (UAVs), and in task allocation in robotic swarm systems, should be intensified.

The potential for the Ant Colony Optimization algorithm in the future is wide and varied; as AI and advanced computing power develop, it promises to play an even more significant part in complex optimization challenges across the scientific and engineering disciplines.

REFERENCES

[1] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, 2005.
[2] M. Dorigo, *Optimization, Learning and Natural Algorithms*, Ph.D. dissertation, Politecnico di Milano, Milan, Italy, 1992.
[3] Y. Zheng *et al*., "Ant colony optimization for accelerated pathway identification," *Processes*, MDPI, 2025.
[4] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, 1997.
[5] L. Zimny *et al*., "Ant colony optimization for parallel test assembly," *PMC Health Sciences*, 2024.
[6] T. Stützle and M. Dorigo, "A short convergence proof for a class of ACO algorithms," *IEEE Transactions on Evolutionary Computation*, 2002.
[7] O. Shalash, "Optimized decentralized swarm communication algorithms for efficient task allocation," *Robotics*, 2024.
[8] W. J. Gutjahr, "A generalizing convergence result for the graph-based ant system," *IEEE Transactions on Evolutionary Computation*, 2002.
[9] M. Alhanjouri and H. AlFarra, "A comparison of ant colony optimization, genetic algorithm, and simulated annealing for TSP," *International Journal of Intelligent Computing Research*, 2015.
[10] B. Santosa, "Ant colony optimization tutorial: Step by step numerical example," 2015.
[11] T. Stützle and M. Dorigo, "ACO algorithms for the traveling salesman problem," 1998.
[12] A. S. B. Shahadat, M. A. H. Akhand, and M. A. S. Kamal, "Visibility adaptation in ant colony optimization for solving traveling salesman problem," *Mathematics*, 2022.
[13] A. S. B. Shahadat, M. A. H. Akhand, and M. A. S. Kamal, "Visibility adaptation in ant colony optimization for solving traveling salesman problem," *Mathematics*, 2022.
[14] I. Khan, M. K. Maiti, and M. Maiti, "Coordinating particle swarm optimization, ant colony optimization and K-opt algorithm for traveling salesman problem," *Communications in Computer and Information Science*, 2017.
[15] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing*, 2015.
[16] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with Levy flight," *IEEE Access*, 2022.
[17] E. Liao and C. Liu, "A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem," *IEEE Access*, 2018.